

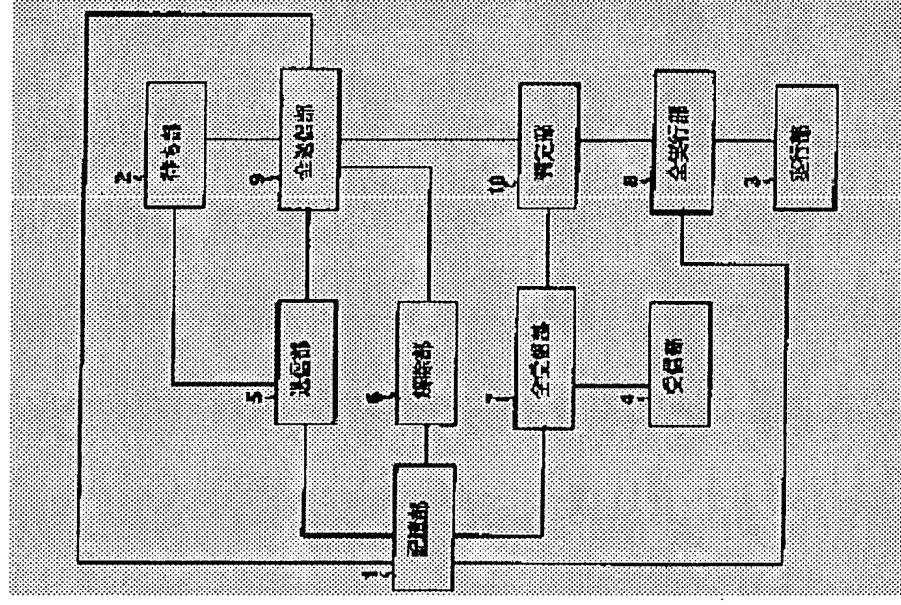
SYNCHRONOUS TYPE MESSAGE COMMUNICATION DEVICE

Patent number: JP5303503
Publication date: 1993-11-16
Inventor: OTO HIDETAKA; others: 01
Applicant: MATSUSHITA ELECTRIC IND CO LTD
Classification:
- international: G06F9/46
- european:
Application number: JP19920109964 19920428
Priority number(s):

Abstract of JP5303503

PURPOSE: To decrease the frequency of the dispatch of a receiving task and reduce the overhead by taking all transmitted messages out at the same time when the receiving task is in dispatch and sending them back to a transmitting task at the same time.

CONSTITUTION: This synchronous type message communication device is equipped with a storage part 1, a wait part 2, an execution part 3, a reception part 4, a transmission part 5, a resetting part 6, an all-reception part 7, an all-execution part 8, an all-transmission part 9, and a decision part 10. When there are plural transmitted messages to the reception-destination task, the all-reception part 7 takes all the messages out at the same time when the reception-destination task is in dispatch, the all-execution part 8 executes the reception-destination task for all the transmission sources, and the all-transmission part 9 sends them back to all the transmission sources. The resetting part 6 deletes all transmitted and received information from the storage part 1 including the identifier of its task and when its task is a receiving task, the decision part 10 calls the all-execution part 8 and then calls the all-transmission part 9.



| | | |
|----------------------------|----------------------------|----------|
| (51) Int. Cl. G06F 9/46 | (52) 識別記号 340 B 8120-5B | (53) F I |
| 審査請求 未請求 請求項の数 2 (全 7 頁) | | |

| | |
|-----------------------------|---|
| (21) 出願番号 特願平4-109964 | (71) 出願人 000005821 松下電器産業株式会社 大阪府門真市大字門真1006番地 大戸 英徳 産業株式会社内 |
| (22) 出願日 平成4年(1992)4月28日 | (72) 発明者 佐々木 裕之 大阪府門真市大字門真1006番地 松下電器産業株式会社内 |
| | (74) 代理人 弁理士 中島 司朗 |

(54) 【発明の名称】 同期型メッセージ通信装置

(57) 【要約】

【目的】 受信タスクのディスプレイバッチの回数を少なくし、ディスプレイバッチのオーバーヘッドを減少することができ、同期型メッセージ通信装置を提供する。

【構成】 受信タスクに対して複数の送信メッセージが送信されていた場合、全受信部7により受信タスクのディスプレイバッチ時に全メッセージを同時に取り出し、全実行部8により全送信元分だけ受信タスクを実行し、かつ全送信部9により全送信元への返信を行う。したがって、受信タスクのディスプレイバッチの回数が少なくなく、ディスプレイバッチのオーバーヘッドを減少させることができる。

【特許請求の範囲】

【請求項1】 受信タスクに対して複数の送信メッセージが送信されていた場合、受信タスクのディスプレイバッチ時に全ての前記送信メッセージを同時に取り出し、かつ送信タスクへの返信を同時に行なってタスク待ち状態を同時に解除する構成としたことを特徴とする同期型メッセージ通信装置。

【請求項2】 受信タスク識別子と送信タスク識別子の組である送受信情報を記憶する記憶部と、タスク再開ポイントを指定して自タスクをタスク待ち状態へ移行する待ち部と、

タスクの実行を行なう実行部と、
タスクを指定してメッセージの受信を行なう受信部と、
自タスクの識別子を送信タスク識別子とし、かつ送信タスクの識別子を受信タスク識別子として作った送受信情報を、前記記憶部に記憶させた後、メッセージを送信する送信部と、

前記記憶部から自タスクの識別子を含む送受信情報を全で削除する解除部と、

前記記憶部の中で受信タスク識別子が自タスクの識別子と等しい全ての送受信情報から送信タスク識別子を全て取り出し、個々の送信タスク識別子に対応する分だけ前記受信部を呼び出す全受信部と、

前記記憶部の中で受信タスク識別子が自タスクの識別子と等しい全ての送受信情報から送信タスク識別子を全て取り出し、個々の送信タスク識別子に対応する分だけ前記実行部を呼び出す全実行部と、

前記記憶部の中で受信タスク識別子が自タスクの識別子と等しい全ての送受信情報から送信タスク識別子を全て取り出し、個々の送信タスク識別子に対応する分だけ前記送信部を呼び出した後、前記待ち部を呼び出す全送信部と、

自タスクが受信タスク状態であれば、前記全実行部を呼び出した後、前記全送信部を呼び出す判定部とを備えたことを特徴とする同期型メッセージ通信装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、計算機システムにおける同期型メッセージ通信装置に関するものである。

【0002】

【従来の技術】 近年、計算機システムで同期型メッセージ通信機能を使用する機会が多く、ユーザに対する実行サービスの応答性や公平性が重要な課題となっている。

図4は従来の同期型メッセージ通信装置の構成図で、この同期型メッセージ通信装置は、待ち部41と、実行部42と、受信部43と、送信部44と、判定部45とを備えている。待ち部41は、タスク開始ポイントを指定して自タスクをタスク待ち状態へ移行する。実行部42は、受信タスクの実行を行なう。受信部43は、送信タスクを指定してメッセージの受信を行なう。送信部44

は、メッセージを送信した後、待ち部41を起動する。判定部45は、自タスクが送信タスクからのメッセージを受信した受信タスクであれば、実行部42を起動した後、受信部43を起動する。

【0003】 この従来の同期型メッセージ通信装置の送信タスク側としての動作を説明する。いま、判定部45の状態変数は送信タスク状態となっている。

(1) まず送信部44は、送信タスクの要求を受けてメッセージを送信し、タスク再開ポイントをとして受信部43を渡して待ち部41を起動する。

(2) これにより待ち部41は、受信部43をタスク再開ポイントとしてタスク待ち状態へ移行する。

【0004】 (3) スケジューラによりディスプレイバッチされて受信部43が起動される。

(4) これにより受信部43は、メッセージを受信した後、判定部45を起動する。

(5) これにより判定部45は、状態変数が送信タスク状態であるため実行を終了する。

【0005】 以上で送信タスク側としての動作が終了する。次に従来の同期型メッセージ通信装置の受信タスク側としての動作を説明する。いま、判定部45の状態変数は受信タスク状態となっている。

(6) まず受信部43は、送信タスクから送られたメッセージを受信した後、判定部45を起動する。
【0006】 (7) これにより判定部45は、状態変数が受信タスク状態であるため実行部42を起動する。

(8) これにより実行部42は、受信タスクを実行する。

(9) 次に判定部45は、送信部44を起動する。

(10) これにより送信部44は、送信タスクへ返信メッセージを送った後、待ち部41を起動する。

【0007】 (11) これにより待ち部41は、受信部43をタスク再開ポイントとしてタスク待ち状態へ移行する。

以上で受信タスク側としての動作が終了する。次に上記従来の同期型メッセージ通信装置の具体的な動作の一例について、図5のタイミングチャートを参照しながら説明する。

【0008】 いま、4つのタスクA、B、C、Dがあり、タスクA、タスクB、タスクCはそれぞれ受信タスクDにメッセージを送信する送信タスクであるものとする。また、単位時間をいすると、各タスクは3つのオンタムで動作し、タスク優先度はA=B=C>Dの関係にあるものとする。また、タスクDはタスクのディスプレイバッチのオーバーヘッド/2を含め、各送信タスクからのメッセージに対する全ての処理をt以下で処理するものとする。また、判定部45の状態変数をstateとし、state=clinetのとき送信タスク状態、state=serverのとき受信タスク状態であるとする。つまり、タスクA、タスクB、タスクCの状態変数state=clientであ

り、タスクDの状態変数state=serverであるとする。

【0009】(21) 先ずタスクAがディスプレイバッサ

れ、時刻0から時刻3 tの間に送信先であるタスクDとメッセージを渡して送信部4 4を呼び出す。

(22) これにより送信部4 4は、タスクAの要求を受けてタスクDにメッセージを送信し、待ち部4 1を起動する。

(23) これにより待ち部4 1は、タスクAの受信部4 3をタスク再開ポイントとしてタスク待ち状態へ移行する。

【0010】(24) タスクB、タスクCについて上配(21)～(23)と同様の動作を行なう。

(25) 次にタスクDがディスプレイバッサされ、受信部4 3を呼び出す。

(26) 次に判定部4 5は、state=serverであるため実行部4 2を起動する。

(27) これにより実行部4 2は、受信タスクを実行する。

【0011】(28) 次に判定部4 5は、送信部4 4を起動する。

(29) これにより送信部4 4は、タスクAへ送信メッセージを送った後、待ち部4 1を起動する。

(30) これにより待ち部4 1は、受信部4 3をタスク再開ポイントとしてタスクDをタスク待ち状態へ移行する。上記(25)からこの(30)までの動作は時刻9 tから10 tまでの間に行われる。

【0012】(31) 次にタスクAがディスプレイバッサされ、受信部4 3が起動される。

(32) これにより受信部4 3は、メッセージを受信した後、判定部4 5を起動する。

(33) これにより判定部4 5は、state=clientであるため同期型メッセージ通信装置の実行を終了する。この後もタスクAは10 tから13 tの間動作を続ける。

【0013】(34) タスクB、タスクCについても上記(25)～(33)と同様の動作を行なう。このとき、図5から明らかなように、タスクDは合計3回(送信タスク数だけ)ディスプレイバッサされ、タスクA、タスクB、タスクCからの同期型メッセージ通信の送受信完了までにタスクDのディスプレイバッサオーバーヘッドは合計3 t/2となる。

【0014】(発明が解決しようとする課題) 上記従来の同期型メッセージ通信装置では、1つの受信タスクに対して複数の送信タスクが順番にメッセージ送信を行なう場合には、受信タスクの処理時間が短いつきでも送信タスク数だけディスプレイバッサが発生し、ディスプレイバッサのオーバーヘッドが大きいという問題があった。

【0015】本発明はかかる事情に鑑みて成されたものであり、受信タスクのディスプレイバッサの回数を少なくし、ディスプレイバッサのオーバーヘッドを減少することができる同

期別子と自タスクの期別子との組である送受信情報を数定する。

【0020】(42) さらに送信部5は、タスク再開ポイントとして全受信部7を渡して待ち部2を起動する。

(43) これにより待ち部2は、全受信部7をタスク再開ポイントとしてタスク待ち状態へ移行する。

(44) スケジューラによりディスプレイバッサされて全受信部7が起動される。

【0021】(45) これにより全受信部7は、メッセージを受信した後、記数部1を起動する。

(46) 次に判定部10は、状態変数が送信タスク状態であるため実行を終了する。以上で送信タスク側としての動作が終了する。

【0022】次に、受信タスク側としての動作について述べる。いま、判定部10の状態変数は受信タスク状態となっている。

(47) 先ず全受信部7は、記数部1の中で自タスクの期別子と等しい受信タスク期別子を持つ送受信情報から送信タスク期別子を全て取り出し、それらに対応する全ての送信タスクに対して各送信タスク毎に受信部4を起動する。

【0023】(48) これにより受信部4は、指定された送信タスクからのメッセージを受信する。この動作は、上記(47)で指定された回数だけ繰り返される。

(49) 次に全受信部7は、判定部10を起動する。

(50) これにより判定部10は、状態変数が受信タスク状態であるため全実行部8を起動する。

【0024】(51) これにより全実行部8は、記数部1の中で自タスクの期別子と等しい受信タスク期別子を持つ送受信情報から送信タスク期別子を全て取り出し、それらに対応する全ての送信タスクに対して各送信タスク毎に実行部3を起動する。

(52) これにより実行部3は、受信タスクを実行する。この動作は、上記(51)で指定された回数だけ繰り返される。

【0025】(53) 次に判定部10は、全送信部9を起動する。

(54) これにより全送信部9は、記数部1の中で自タスクの期別子と等しい受信タスク期別子を持つ送受信情報から送信タスク期別子を全て取り出し、それらに対応する全ての送信タスクに対して各送信タスク毎に送信部5を起動する。

(55) これにより送信部5は、送信タスクへの送信を行なう。この動作は、上記(54)で指定された回数だけ繰り返される。

【0026】(56) 次に全送信部9は、解除部6を起動する。

(57) これにより解除部6は、自タスクの期別子を含む送受信情報を記数部1の中から全て削除する。

(58) 次に全送信部9は、待ち部2を起動する。

(59) これにより待ち部2は、全受信部7をタスク再開ポイントとしてタスク待ち状態へ移行する。

【0027】以上で受信タスク側としての動作が終了する。次に上記同期型メッセージ通信装置の具体的な動作の一例について、タスクのタイミングチャートである図2および記数部1のデータ構造の説明図である図3を参照しながら説明する。いま、4つのタスクA、B、C、Dがあり、各々の期別子をTa、Tb、Tc、Tdとする。また、タスクA、タスクB、タスクCはそれぞれ受信タスクDにメッセージを送信する送信タスクであるものとする。また、単位時間をとすると、各タスクは3 tの周期で動作し、タスク優先度はA=B=C>Dの順にあるものとする。また、タスクDはタスクのディスプレイバッサのオーバーヘッドt/2を含め、各送信タスクからのメッセージに対する全ての処理をt以下で処理するものとする。また、判定部10の参照する状態変数はstateとし、state=clientのとき送信タスク状態、state=serverのとき受信タスク状態であるとする。つまり、タスクA、タスクB、タスクCの状態変数state=clientであり、タスクDの状態変数state=serverであるとする。

【0028】(61) 先ずタスクAがディスプレイバッサされ、時刻0から時刻3 tの間に送信先であるタスクDとメッセージを渡して送信部5を呼び出す。

(62) これにより送信部5は、タスクAの要求を受けてタスクDにメッセージを送信し、記数部1に受信タスクであるタスクDとタスクAとの期別子の組(Td, Ta)を設定する。

【0029】(63) さらに送信部5は、待ち部2を起動する。

(64) これにより待ち部2は、タスクAの全受信部7をタスク再開ポイントとしてタスク待ち状態へ移行する。

(65) タスクB、タスクCについても上記(61)～(64)と同様の動作を行なう。

【0030】(66) 次にタスクDがディスプレイバッサされ、全受信部7を呼び出す。

(67) これにより全受信部7は、記数部1の中で自タスクである受信タスク期別子と送信タスク期別子との組である送受信情報(Td, Ta)、(Td, Tb)、(Td, Tc)を全て取り出し、各送信タスク毎に受信部4を起動する。

(68) これにより受信部4は、指定されたタスクからのメッセージを受信する。この動作は、タスクAからタスクCまで繰り返される。

【0031】(69) 次に全受信部7は、判定部10を起動する。

(70) これにより判定部10は、state=serverであるため全実行部8を起動する。

(71) これにより全実行部8は、記数部1の中で自タスクである受信タスク期別子と送信タスク期別子との組

を全て取り出し、各送信タスク毎に受信部4を起動する。

(72) これにより受信部4は、指定されたタスクからのメッセージを受信する。この動作は、タスクAからタスクCまで繰り返される。

(73) 次に全送信部9は、待ち部2を起動する。

(74) これにより待ち部2は、全受信部7をタスク再開ポイントとしてタスク待ち状態へ移行する。

(75) 以上で受信タスク側としての動作が終了する。

である送受信情報 (Td, Ta), (Td, Tb), (Td, Tc) を全て取り出し、各送信タスク毎に実行部3を起動する。

[0032] (72) これにより実行部3は、受信タスクを実行する。この動作はタスクAからタスクCまで繰り返される。

(73) 次に判定部10は、全送信部9を起動する。

(74) これにより全送信部9は、記憶部1の中で自タスクである受信タスク識別子と送信タスク識別子との組である送受信情報 (Td, Ta), (Td, Tb), (Td, Tc) を全て取り出し、各送信タスク毎に送信部5を起動する。

[0033] (75) これにより送信部5は、送信タスクへの返信を行なう。この動作はタスクAからタスクCまで繰り返される。

(76) 次に全送信部9は、解除部6を起動する。

(77) これにより解除部6は、記憶部1の中から送受信情報 (Td, Ta), (Td, Tb), (Td, Tc) を削除する。

[0034] (78) 次に全送信部9は、待ち部2を起動する。

(79) これにより待ち部2は、全受信部7をタスク再帰ポイントとしてタスク待ち状態へ移行する。上記 (6) からこの (79) までの動作は、時刻9 tから11 tの間に行われる。

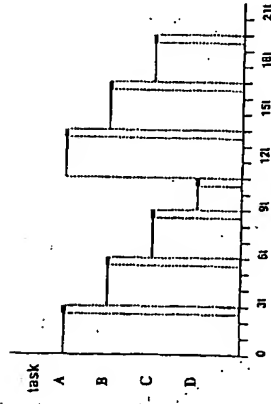
(80) 次にタスクAがダイスバッチされ、全受信部7が起動される。

[0035] (81) これにより全受信部7は、メッセージを受信した後、判定部10を起動する。

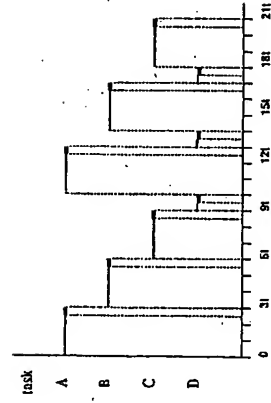
(82) これにより判定部10は、state-clientであるため同期型メッセージ通信装置の実行を終了する。この後もタスクAは11 tから14 tの間動作を続ける。

[0036] (83) 次に、タスクB、タスクCについても上記 (80) ~ (82) と同様の動作を行なう。図

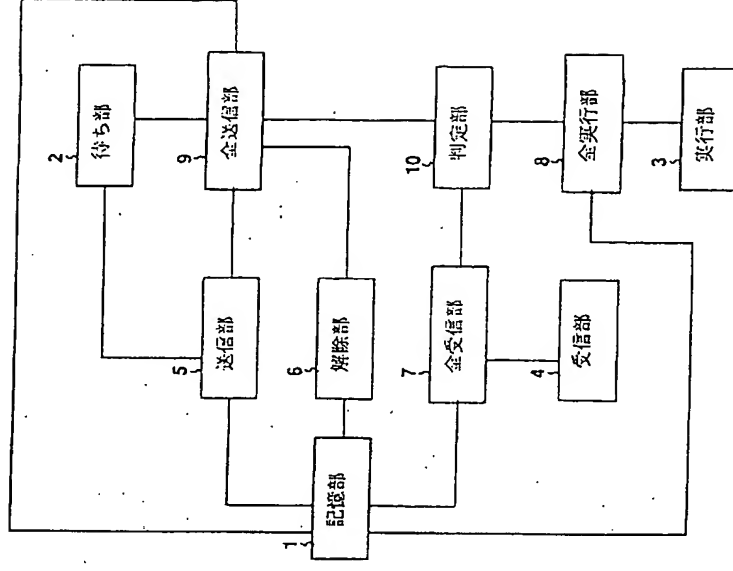
[図2]



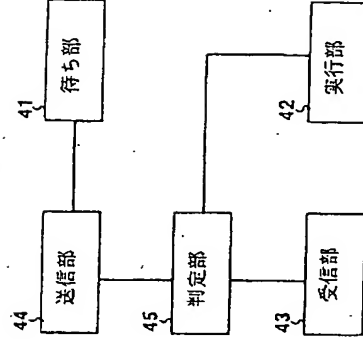
[図5]



[図1]



[図4]



【図3】

